

2주차

기본 입출력
`printf()` / `scanf_s()`

연산자, 제어흐름

scanf_s

```
int main(void) {
```

```
    int num;
```

```
    scanf_s( "%d", &num );
```

변수의 주소

형식 문자

```
    return 0;
```

```
}
```

scanf_s 형식문자

종류	자료형	지정자	설명	종류	자료형	지정자	설명
정수	10진수	signed short	%hi	실수	float	%f	10진수 실수 입력(4바이트 크기)
		unsigned short	%hu		double	%lf	10진수 double형 실수 입력(8바이트 크기)
		signed int	%d		long double	%llf	10진수 long double형 실수 입력(12바이트 크기)
		unsigned int	%u		float, double	%e	부동 소수점 e(소문자) 형식의 실수 입력
		signed long	%li		float, double	%E	부동 소수점 E(대문자) 형식의 실수 입력
		unsigned long	%lu		float, double	%g	%e와 %g 중에서 길이가 짧은 쪽을 입력
		signed long long	%lli		float, double	%G	%e와 %g 중에서 길이가 긴 쪽을 입력
		unsigned long long	%llu	문자	char	%c	문자 데이터 입력
	8진수	signed short	%ho	문자열	char*, char[]	%s	문자열 데이터 입력
		unsigned int	%o	포인터	void*	%p	포인터 주소값 입력
	16진수	signed short	%hx				
		unsigned int	%x				
		signed short	%hX				
		unsigned int	%X				



자주 사용

실수 입력

```
int main(void) {  
    float f;  
    double d;  
  
    scanf_s( "%lf", &d);  
    printf( "%lf", d);  
}
```

문자 입력

```
int main(void) {  
    int num;  
    char c;  
  
    scanf_s("%c", &c, sizeof(c));  
    printf("%c", c);  
}
```

scanf() 엔터키의 입력

```
int main(void){  
    char a, b;  
    printf("첫번째 문자 입력\n");  
    scanf_s("%c", &a, 1);  
    printf("두번째 문자 입력\n");  
    scanf_s("%c", &b, 1);  
    printf("%c, %c\n", a, b); // b는 LF(Line Feed) 공백문자  
}
```

scanf() 엔터키의 입력

```
int main(void) {  
    char a, b;  
    printf("첫번째 문자 입력\n");  
    scanf_s("%c", &a, 1);  
    printf("두번째 문자 입력\n");  
    rewind(stdin);  
    scanf_s("%c", &b, 1);  
  
    printf("첫번째 문자 : %c, 두번째 문자: %c\n", a, b);  
}
```

문자열 입력

```
int main(void) {  
    char str[10];  
  
    scanf_s("%s", str, sizeof(str));  
    printf("%s", str);  
}
```


scanf() 형식 연속입력

```
int main(void){  
    int a, b;  
    char c;  
    printf("1 + 2 형태로 입력\n");  
    scanf_s("%d %c %d", &a, &c, 1, &b);  
    printf("%d, %c, %d\n", a, c, b);  
}
```

printf

크게 2가지의 형태로 출력을 표현

```
int main(void) {  
    printf( "%d\n" , 10);  
}
```

1. 형식 문자

2. 제어 문자

1. 형식문자



자주 사용

종류	자료형	지정자	설명
정수	10진수	%d	부호 있는 정수 출력(4바이트 크기)
		%ld	부호 있는 long형 정수 출력(4바이트 크기)
		%u	부호 없는 정수 출력(4바이트 크기)
	8진수	%o	8진수 정수 출력
		%lo	long형 8진수 정수 출력
	16진수	%x	16진수 소문자 출력
		%lx	long형 16진수 소문자 출력
		%X	16진수 대문자 출력
		%lX	long형 16진수 대문자 출력
실수	float, double	%f	10진수 실수 출력(4바이트 크기)
		%e	부동 소수점 e(소문자) 형식의 실수 출력
		%E	부동 소수점 E(대문자) 형식의 실수 출력
		%g	%e와 %g 중에서 길이가 짧은 쪽 출력
		%G	%e와 %g 중에서 길이가 긴 쪽 출력
문자	char	%c	문자 데이터 출력
문자열	char*, char[]	%s	문자열 데이터 출력
포인터	void*	%p	포인터 주소값 출력

2. 제어문자

제어 문자	명칭	설명
\a	bell	뽁~ 소리를 냄
\b	back space	왼쪽으로 한 칸 이동
\f	form feed	프린터에서 한 페이지 이동
\n	line feed	줄을 바꿈
\r	carriage return	커서를 1열로 이동하여 덮어쓰움
\t	tab	Tab 키와 동일한 기능
\\	back slash	역슬래시(\) 출력
\'	apostrophe	작은따옴표(') 출력
\"	quote	큰따옴표(") 출력

제어문자 예시

```
int main(void) {  
    char str[10];  
  
    scanf_s( "%s", str, sizeof(str));  
    printf( "input string: %s\n", str);  
}
```

3. 형식문자 정렬

[전체 자릿수 지정]

%(전체 자릿수)d

[빈칸을 0으로 채운 전체 자릿수 지정]

%0(전체 자릿수)d

%d



3 5 8



정수형 기본 자릿수만큼 출력

%5d



3 5 8



5개의 자리 확보한 채 오른쪽 정렬로 출력

%05d



0 0 3 5 8



5개의 자리 확보, 빈칸은 0으로 채운 채
오른쪽 정렬로 출력

정렬 예시

```
int main(void) {  
    float f = 3.14;  
  
    printf( "%20f ", f );  
    printf( "%-20f ", f );  
}
```

3. 형식문자 자릿수 정렬

[소수점 포함 전체 자릿수 지정]

%(전체 자릿수).(소수점 이하 자릿수)f

[왼쪽 정렬+소수점 포함 전체 자릿수 지정]

%-(전체 자릿수).(소수점 이하 자릿수)f

[소수점 이하 자릿수 지정]

%(소수점 이하 자릿수)f

3. 형식문자 자릿수 정렬

%f	➡	<table><tr><td>3</td><td>5</td><td>8</td><td>.</td><td>1</td><td>2</td><td>3</td><td>0</td><td>0</td><td>0</td></tr></table>	3	5	8	.	1	2	3	0	0	0	← 실수형 기본 자릿수인 소수 여섯째 자리까지 무조건 출력
3	5	8	.	1	2	3	0	0	0				
%2f	➡	<table><tr><td>3</td><td>5</td><td>8</td><td>.</td><td>1</td><td>2</td></tr></table>	3	5	8	.	1	2	← 소수 둘째 자리까지 출력. 셋째 자리에서 반올림. 숫자는 오른쪽 정렬				
3	5	8	.	1	2								
%6.1f	➡	<table><tr><td></td><td>3</td><td>5</td><td>8</td><td>.</td><td>1</td></tr></table>		3	5	8	.	1	← 소수 첫째 자리까지 출력. 전체 6개의 자리 확보. 둘째 자리에서 반올림				
	3	5	8	.	1								
%-6.1f	➡	<table><tr><td>3</td><td>5</td><td>8</td><td>.</td><td>1</td><td></td></tr></table>	3	5	8	.	1		← 소수 첫째 자리까지 출력. 전체 6개의 자리 확보. 둘째 자리에서 반올림. 숫자는 왼쪽 정렬				
3	5	8	.	1									
%7.3f	➡	<table><tr><td>3</td><td>5</td><td>8</td><td>.</td><td>1</td><td>2</td><td>0</td></tr></table>	3	5	8	.	1	2	0	← 소수 셋째 자리까지 출력. 전체 7개의 자리 확보. 오른쪽 빈칸은 숫자 0으로 채움			
3	5	8	.	1	2	0							
%010.3f	➡	<table><tr><td>0</td><td>0</td><td>0</td><td>3</td><td>5</td><td>8</td><td>.</td><td>1</td><td>2</td><td>3</td></tr></table>	0	0	0	3	5	8	.	1	2	3	← 소수 셋째 자리까지 출력. 전체 10개의 자리 확보. 빈칸은 0으로 채움
0	0	0	3	5	8	.	1	2	3				

자릿수 예시

```
int main(void) {  
    float pi = 3.14159;  
  
    printf( "%f\n", pi);  
    printf( "%g\n", pi);  
    printf( "%.3f\n", pi);  
    printf( "%10.3f\n", pi);  
    printf( "%010.3f\n", pi);  
}
```

Puts 함수

```
int main(void) {  
    float pi = 3.14159;  
  
    //printf("자릿수 예시\n");  
    puts("자릿수 예시");  
    printf("%f\n", pi);  
    printf("%g\n", pi);  
    printf("%.3f\n", pi);  
    printf("%10.3f\n", pi);  
    printf("%010.3f\n", pi);  
}
```

산술 연산자

+

-


*

/

%

```
int main(void) {  
    int n1 = 5;  
    int n2 = 7;  
    printf("두 수의 합: %d\n", n1 + n2);  
}
```

결과 값



```
int main(void) {  
    int n1, n2;  
  
    puts("두 정수 입력");  
    scanf_s("%d", &n1);  
    rewind(stdin);  
    scanf_s("%d", &n2);  
    printf("%d + %d = %d\n", n1, n2, n1 + n2);  
    printf("%d - %d = %d\n", n1, n2, n1 - n2);  
    printf("%d * %d = %d\n", n1, n2, n1 * n2);  
    printf("%d / %d 몫 = %d\n", n1, n2, n1 / n2);  
    printf("%d / %d 나머지 = %d\n", n1, n2, n1 % n2);  
}
```

```
int main(void) {  
    const int min = 60; // 1분은 60초  
    int seconds; // 초 입력
```

Microsoft Visual Studio 디버그 콘솔

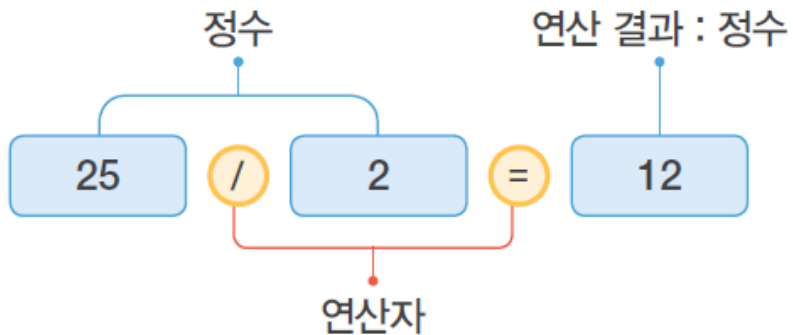
```
초를 입력하세요:  
1204  
1204초는 20분 4초 입니다.
```

```
}
```

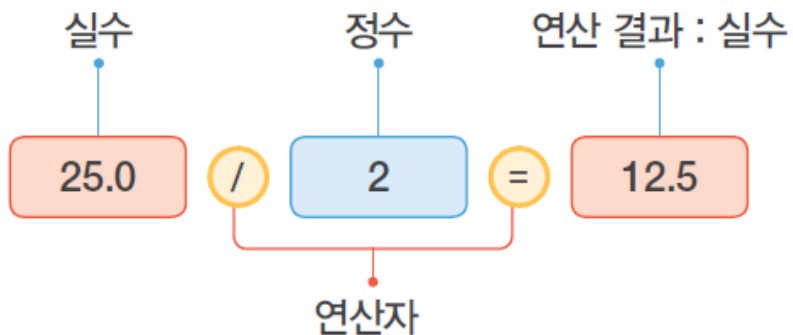
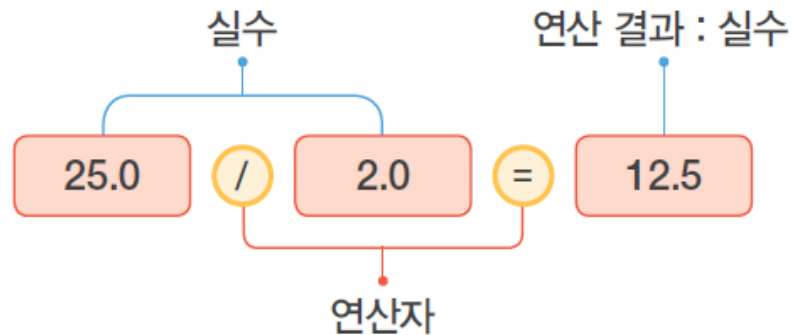
정수의 나눗셈 실수 연산하기

```
int main(void) {  
    int n1 = 10;  
    int n2 = 3;  
  
    printf("두 정수를 나눈 실수값 : %f\n",  
        (float)n1 / n2);  
    //float / int -> float  
}
```

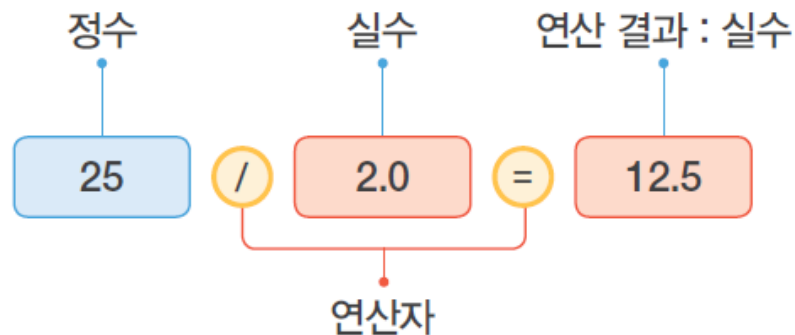

자료형과 연산결과



자료형이 같은 경우의 연산 결과




자료형이 다른 경우의 연산 결과



```
int main(void) {
```

```
    int cm;
```

 Microsoft Visual Studio 디버그 콘솔

```
cm단위를 입력하세요  
35  
35cm 는 0.35m 입니다.
```

```
int main(void) {  
  
    float r;  
    const float PI = 3.14;
```

 Microsoft Visual Studio 디버그 콘솔

원의 반지름을 입력하세요

5.5

입력된 반지름 : 5.5, 원의 넓이 : 34.54

대입 / 증감 연산자

```
int main(void) {  
    int num = 3;  
  
    num = num + 10; // 단순 대입  
    num += 10; // 복합 대입, 위와 동일한 식  
  
    printf("%d", num++); // 변수 사용 후 1증가  
    printf("%d", --num); // 변수 사용 전 1감소  
    printf("%d", num);  
}
```

관계 연산자

종류	사용 예	설명	연산 결과 예시	결과값
<code>==</code>	<code>a == b</code>	a와 b는 같음	<code>5 == 8</code>	0
<code>!=</code>	<code>a != b</code>	a와 b는 같지 않음	<code>5 != 8</code>	1
<code>></code>	<code>a > b</code>	a는 b보다 큼	<code>5 > 8</code>	0
<code><</code>	<code>a < b</code>	a는 b보다 작음	<code>5 < 8</code>	1
<code>>=</code>	<code>a >= b</code>	a는 b보다 크거나 같음	<code>5 >= 8</code>	0
<code><=</code>	<code>a <= b</code>	a는 b보다 작거나 같음	<code>5 <= 8</code>	1

논리 연산자

종류	사용 예	설명
!(NOT)	!a	a가 참임을 부정하여 거짓, a가 거짓임을 부정하여 참
&&(AND)	a && b	a와 b가 모두 참일 경우 참, 하나라도 거짓이면 거짓
(OR)	a b	a와 b 중에서 하나라도 참이면 참, 모두 거짓이면 거짓

a	b	a && b	a b	!a	!b
0	0	0	0	1	1
1	0	0	1	0	1
0	1	0	1	1	0
1	1	1	1	0	0

알파벳 체크

041)	065	A	089	Y	113	q
042	*	066	B	090	Z	114	r
043	+	067	C	091	[115	s
044	,	068	D	092	\	116	t
045	-	069	E	093]	117	u
046	.	070	F	094	^	118	v
047	/	071	G	095	_	119	w
048	0	072	H	096	`	120	x
049	1	073	I	097	a	121	y
050	2	074	J	098	b	122	z
051	3	075	K	099	c	123	{
052	4	076	L	100	d	124	
053	5	077	M	101	e	125	}
054	6	078	N	102	f	126	~
055	7	079	O	103	g		

알파벳 체크

```
int main(void) {  
    char a; // 알파벳 체크  
  
    puts("알파벳을 입력하세요");  
    scanf_s("%c", &a, 1);  
    printf("입력된 알파벳은 %c, 알파벳 체크 결과 (참1) (거짓0) : %d",  
        a, (65 <= a && a <= 90) || (97 <= a && a <= 122));  
}
```



```
int main(void) {  
    int n; // 홀수 짝수 체크
```

Microsoft Visual Studio 디버그 콘솔

정수를 입력하세요

13

입력된 정수는 13 홀수(0) 짝수(1) 여부 : 0

```
}
```

scanf -> scanf_s 보안 취약점 (BOF)

```
#include <stdio.h>
```

```
#pragma warning(disable: 4996)
```

```
int main(void) {
```

```
    int num = 0;
```

```
    char c[10];
```

```
    scanf("%s", &c);
```

```
    printf("c: %p, num: %p\n", &c, &num);
```

비트연산자

A	B	A & B	A B	A ^ B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

연산자	명칭	설명
&	비트 논리곱(AND)	둘 다 1이면 1이다.
	비트 논리합(OR)	둘 중 하나만 1이면 1이다.
^	비트 배타적 논리합(XOR)	둘이 같으면 0, 둘이 다르면 1이다.
~	비트 부정	1은 0으로, 0은 1로 변경한다.
<<	비트 왼쪽 시프트(이동)	비트를 왼쪽으로 시프트(이동)한다.
>>	비트 오른쪽 시프트(이동)	비트를 오른쪽으로 시프트(이동)한다.

```
int main(void)
{
    int a = 0b01101011; // (10진수 107)
    int b = 22; // 0b00010110 (2진수 00010110)
    // a : 01101011
    // b : 00010110

    printf( "%d\n", a & b);
    printf( "%d\n", a | b);
    printf( "%d\n", a ^ b);
}
```

비트연산자

《 | 비트 왼쪽 시프트(이동)



비트연산자

» | 비트 오른쪽 시프트(이동)



```
int main(void) {  
    int a = 1;  
    int b = 0b1111;  
    printf( "%d\n", a << 1);  
    printf( "%d\n", a << 2);  
    printf( "%d\n", a << 3);  
    printf( "b : %d\n", b);  
    printf( "%d\n", b >> 1);  
    printf( "%d\n", b >> 2);  
    printf( "%d\n", b >> 3);  
}
```

비트연산자를 이용한 2진수변환

```
int main(void) {  
    int num = 0;  
  
    scanf_s( "%d", &num);  
  
    printf( "%d", (num >> 3) & 1);  
    printf( "%d", (num >> 2) & 1);  
    printf( "%d", (num >> 1) & 1);  
    printf( "%d", num & 1);  
}
```


기본적인 if문

if(조건식)
실행할 문장;

```
int main(void) {  
    int n = 10;  
  
    if (n > 0) {  
        puts("n은 0보다 크다.");  
    }  
}
```

if(조건식)
실행할 문장;

블록없는 if문

```
int main(void){  
    int n = 0;  
  
    if (n > 10)  
        puts("첫번째 줄 코드");  
        puts("두번째 줄 코드");  
    puts("마지막 줄 코드");  
}
```

Microsoft Visual Studio 디버그 콘솔

두번째 줄 코드
마지막 줄 코드

If문 + {} 블록

```
int main(void){  
    int n = 0;  
  
    if (n > 10)  
    {  
        puts("첫번째 줄 코드");  
        puts("두번째 줄 코드");  
    }  
    puts("마지막 줄 코드");  
}
```

Microsoft Visual Studio 디버그 콘솔

마지막 줄 코드

if else 문

```
if(조건식)  
    실행할 문장 1;  
else  
    실행할 문장 2;
```

```
int main(void) {  
    int n = 10;  
  
    if (n % 2 == 0) {  
        puts("n은 짝수");  
    }  
    else {  
        puts("n은 홀수");  
    }  
}
```

if else 문의 중첩

```
int main(void) {  
    int n;  
    puts("정수를 입력하세요.");  
    scanf_s("%d", &n);  
  
    if (n > 0) {  
        puts("n은 양수입니다.");  
    }  
    else if (n < 0) {  
        puts("n은 음수입니다.");  
    }  
    else {  
        puts("n은 0 입니다.");  
    }  
}
```

if 문의 동작 순서

```
int main(void){  
    int n = 200;  
  
    if (n > 0) {  
        printf("%d는 0보다 크다.\n", n);  
    }  
    else if (n > 50) {  
        printf("%d는 50보다 크다.\n", n);  
    }  
    else if (n > 100) {  
        printf("%d는 100보다 크다.\n", n);  
    }  
    printf("마지막 코드");  
}
```

Microsoft Visual Studio 디버그 콘솔

```
200는 0보다 크다.  
마지막 코드
```

```
int main(void){  
    int n;  
    scanf_s("%d", &n);  
    if (n > 100) {  
        printf("%d는 100보다 크다.\n", n);  
    }  
    else if (n > 50) {  
        printf("%d는 50보다 크다.\n", n);  
    }  
    else if (n > 0) {  
        printf("%d는 0보다 크다.\n", n);  
    }  
    else {  
        printf("%d는 0보다 작다.\n", n);  
    }  
}
```

```
int main(void) {
    int n; // 입력한 성적
    char grade; // 학점
    puts("성적을 입력하세요. (0 ~ 100점 사이)");
```

등급	점수 구간	조건
A학점	90점 이상 100점 이하	입력한 점수가 90 ~ 100점이면 A 학점
B학점	80점 이상 90점 미만	입력한 점수가 80 ~ 89점이면 B 학점
C학점	70점 이상 80점 미만	입력한 점수가 70 ~ 79점이면 C 학점
D학점	60점 이상 70점 미만	입력한 점수가 60 ~ 69점이면 D 학점
F학점	0점 이상 60점 미만	입력한 점수가 0 ~ 60점이면 F 학점

Microsoft Visual Studio 디버그 콘솔

```
성적을 입력하세요. (0 ~ 100점 사이)
85
입력한 성적의 학점은 : B입니다.
```


if문의 조건문 실수

```
int main(void){  
    int n;  
    scanf_s( "%d", &n);  
    if (n = 100) {  
        printf( "만점입니다!\n");  
    }  
}
```

알파벳 대소문자 변환

```
int main(void){
    char a, b=0;
    printf("알파벳 소문자를 입력하세요 \n");
    scanf_s("%c", &a, 1);
    if (a >= 'a' && a <= 'z') {
        b = a - 32; // 'a' : 97 , 'A' : 65
    }
    printf("입력: %c , 변환 후: %c", a, b);
}
```

```
int main(void){  
    char a, b;  
    printf("알파벳 대문자를 입력하세요 \n");  
    C  
    입력: C , 변환 후: c  
}
```

Microsoft Visual Studio 디버그 콘솔

알파벳 대문자를 입력하세요
C
입력: C , 변환 후: c

```
int main(void){  
    char a, b;  
    printf("알파벳을 입력하세요 \n");  
    scanf_s("%c", &a, 1);  
}
```

Microsoft Visual Studio 디버그 콘솔

알파벳을 입력하세요

G

입력: G , 변환 후: g

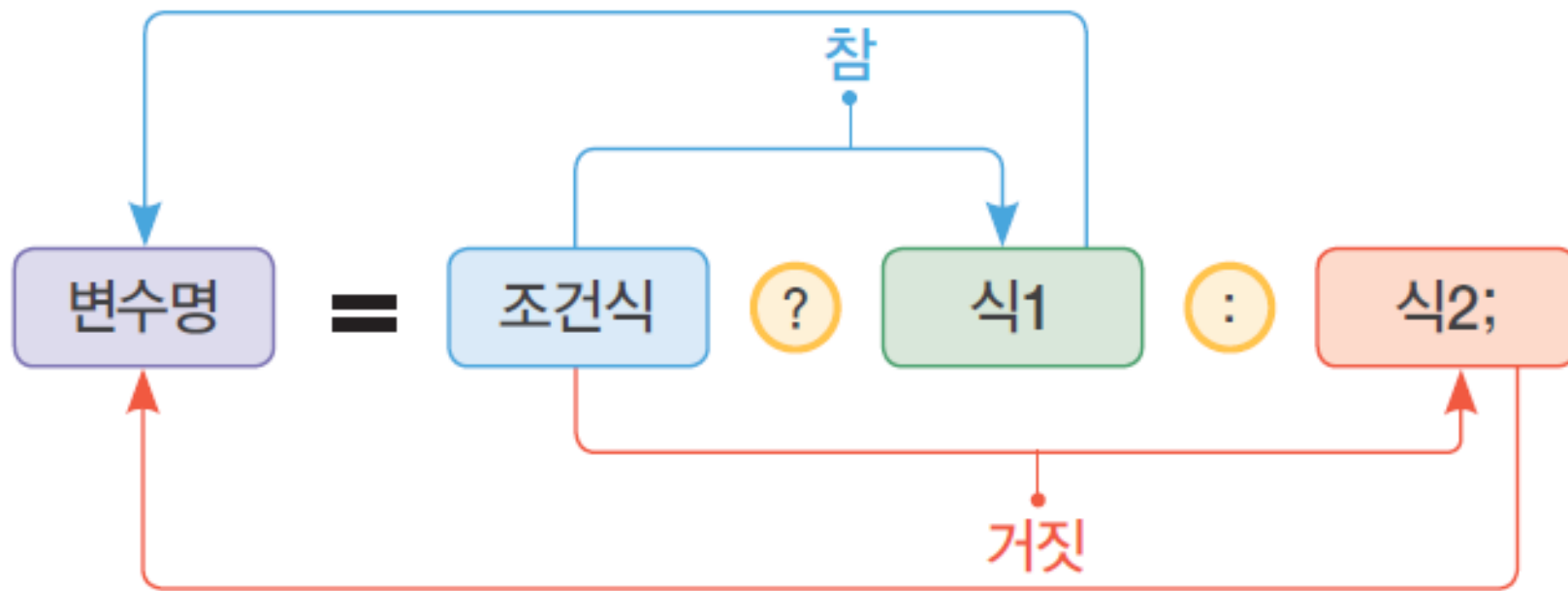
Microsoft Visual Studio 디버그 콘솔

알파벳을 입력하세요

1

알파벳 입력이 아닙니다.

삼항 조건 연산자



절대값 출력

```
int main(void) {  
    int n;  
    int abs; // 절대값 출력  
  
    puts("정수를 입력하세요");  
    scanf_s("%d", &n);  
    abs = n > 0 ? n : n * -1;  
    printf("입력된 숫자는 %d, 절대값 : %d",  
        n, abs);  
}
```

```
int main(void) {  
    char a; // 알파벳 대소문자 변경
```

Microsoft Visual Studio 디버그 콘솔

알파벳을 입력하세요

G

입력된 알파벳은 G, 대소문자 변경 후 : g

```
}
```